

# Improving the performance of a preference-based multi-objective algorithm to optimize food treatment processes

M. R. Ferrández<sup>a\*</sup>, J. L. Redondo<sup>a</sup>, B. Ivorra<sup>b</sup>,  
A. M. Ramos<sup>b</sup>, P. M. Ortigosa<sup>a</sup> and B. Paechter<sup>c</sup>

<sup>a</sup>Department of Computer Science, University of Almería, ceiA3, Ctra. Sacramento, La Cañada de San Urbano, Almería 04120, Spain;

<sup>b</sup>Instituto de Matemática Interdisciplinar (IMI) and Department of Mathematical Analysis and Applied Mathematics, Complutense University of Madrid, Plaza de las Ciencias, 3, Madrid 28040, Spain;

<sup>c</sup>School of Computing, Edinburgh Napier University, Edinburgh, Scotland, UK

*This is an original preprint of an article that has been accepted for publication in **Engineering Optimization**, published by Taylor & Francis.*

## Abstract

This work focuses on the optimization of some high-pressure and temperature food treatments. When dealing with real-life multi-objective optimization problems, the computational cost of evaluating the considered objective functions is usually quite high. Therefore, only a reduced number of iterations is affordable for the optimization algorithm. However, using fewer iterations can lead to inaccurate solutions far from the real Pareto optimal front. In this work, we analyze and compare different mechanisms to improve the convergence of a preference-based multi-objective optimization algorithm called Weighting Achievement Scalarizing Function Genetic Algorithm. The combination of these techniques has been applied for optimizing a particular food treatment process. In particular, the proposed method based on the introduction of an advanced population achieves important improvements in the considered quality indicator measures.

**Keywords:** preference-based multi-objective optimization algorithm; low-cost optimization; food treatment

---

\*CONTACT M. R. Ferrández. Email: mrferrandez@ual.es

# 1 Introduction

Recently, High-Pressure and Temperature (HPT) processes have emerged as a reference technology in the food industry, mainly due to the fact that they enhance the preservation of some good properties (e.g., organoleptic) and reduce the proliferation of damaging micro-organisms without using additives. To produce a processed food within predetermined quality levels, the food engineer must carefully determine the control temperatures and pressures to be applied during the HPT equipment. Nowadays, this challenging task is accomplished by several trial and error tests. However, since the target quality levels depend on the country in which they operate, these tests can lead to a considerable waste of time, money and product. In (Ferrández et al. 2017), we have proposed a decision tool based on solving a multi-objective optimization problem to assist the food engineers in the design of HPT treatments. More precisely, the optimization problem consists in finding the initial and refrigeration temperatures and the pressure profile to be provided to the HPT equipment such that the final enzymatic activity in the food and the maximum temperature reached during the whole process are minimal and the final vitamin activity is maximal. In general, this kind of optimization problems arising in real-life situations usually involve some mathematical models for describing the physical phenomena. Although these models are simplifications of reality, they frequently imply complex computations managing many variables and objectives (see, e.g., Gomez, Ivorra, and Ramos (2011); Ivorra et al. (2013)). In this context, the meta-heuristic algorithms are quite suitable for dealing with their optimization.

In a classical multi-objective optimization problem, formulated as follows:

$$\begin{aligned} & \{\min f_1(\mathbf{x}), \dots, \min f_m(\mathbf{x})\} \\ \text{s.t. } & \mathbf{x} \in S \subseteq \mathbb{R}^n, \end{aligned} \tag{1}$$

the goal is to find a set of *decision vectors*  $\mathbf{x} = (x_1, \dots, x_n)$ , with  $n \in \mathbb{N}$ , belonging to a set called *feasible region*  $S \subseteq \mathbb{R}^n$  and satisfying that the values obtained when evaluating the *objective functions*  $f_1, \dots, f_m : \mathbb{R}^n \rightarrow \mathbb{R}$  at these decision vectors are minimal. However, minimizing all the objectives simultaneously is not always a trivial and feasible task because they frequently confront each other, so the decision vectors of the solution set, known as Pareto optimal set, must be those having the best compromise among the considered objectives.

In this framework, a decision vector  $\mathbf{x}^* \in S$  is said to be *efficient* if and only if there does not exist another feasible vector  $\mathbf{x}$  in  $S$  dominating  $\mathbf{x}^*$ , that is, there does not exist another feasible vector  $\mathbf{x} \in S$  satisfying that  $f_k(\mathbf{x}) \leq f_k(\mathbf{x}^*)$ , for all  $k = 1, \dots, m$ , and  $f_k(\mathbf{x}) < f_k(\mathbf{x}^*)$  for at least one index  $k$  (i. e., none of the objective values can be improved without worsening at least one of the others). The image of these non-dominated efficient vectors in the feasible objective region  $F(S) \subseteq \mathbb{R}^n$  is known as Pareto optimal front.

A decision vector  $\mathbf{x}^*$  in  $S$  is said to be *weakly efficient* if and only if there does not exist another feasible vector  $\mathbf{x} \in S$  such that  $f_k(\mathbf{x}) \leq f_k(\mathbf{x}^*)$ , for all  $k = 1, \dots, m$ .

As said before, the meta-heuristic algorithms are especially prescribed for solving complex problems due to the fact that they do not require any previous information about the objective functions (such as their gradient or Hessian matrix). More precisely, they provide a finite set of points composing a Pareto front approximation as a solution of (1). Among these kinds of algorithms, the most popular subfamily of methods are the so-called evolutionary multi-objective algorithms (EMOA), which are based on iterative procedures that continuously improve the set of approximated solutions bringing them closer to the Pareto front set. When dealing with real-life applications, only a reduced number of iterations may be affordable for the optimization algorithm from the computational point of view due to the high cost of evaluating the objective function. However, using fewer iterations can lead to inaccurate solutions far from the real Pareto optimal front.

A literature review gives us some interesting ideas and methods aimed at enhancing the convergence and, consequently, achieving better results (i.e., solutions closer to the Pareto front) in fewer iterations. For instance, in (Ivorra, Mohammadi, and Ramos 2015), the authors apply a secant technique for improving the initialization of a generic mono-objective optimization algorithm. Other widely-used mechanisms are the local search methods. In particular, the single agent stochastic search (SASS) algorithm (Solis and Wets 1981) was adapted by Lancinskas, Ortigosa, and Žilinskas (2013) to work with multi-objective problems. This extended version called MOSASS showed successful results in (Redondo, Fernández, and Ortigosa 2017; Filatovas et al. 2016), where it is added as part of an evolutionary algorithm.

As done in our previous works (Ferrández et al. 2017; Ferrández et al. 2018), we assume that we are only interested in calculating a part of the Pareto front determined by the food engineer and known as the region of interest. Therefore, we decided to use the EMOA called Weighting Achievement Scalarizing Function Genetic Algorithm (WASF-GA). According to Ruiz, Saborido, and Luque (2015), it provides higher-quality results than other preference-based algorithms when the optimization problem has three or more objectives, as it is our case here.

Now, in this work, we propose some variants of WASF-GA applying different mechanisms in order to achieve a set of solutions that approximate the Pareto front as accurately as possible when considering a low number of iterations. The results obtained with those different WASF-GA variants have been compared in terms of effectiveness using several quality indicator measure methods. More precisely, let  $\Omega$  be the set of all Pareto front approximations, a *unary quality indicator* is a function  $I : \Omega \rightarrow \mathbb{R}$  which assigns each Pareto front approximation set  $PFA \in \Omega$  a real value  $I(PFA)$ .

The article is organized as follows. First, in Section 2, the design of the HPT treatment is formulated as a multi-objective optimization problem. In Section 3, the original WASF-GA is described. Next, in Section 4, different mechanisms to improve WASF-GA are proposed. In Section 5, the computational experiments carried out to compare the original WASF-GA and its considered variants are detailed. Finally, in Section 6, the optimization results are discussed in terms

of some state-of-the-art quality measures.

## 2 Food processing problem

The considered problem consists in finding a set of temperature and pressure configurations for the combined treatments of High-Pressure and Temperature (HPT) that allow us to minimize the enzymatic activity and the maximum temperature and to maximize the vitamin activity, at the same time (Ferrández et al. 2017). It is formulated as follows:

$$\begin{cases} \min & f_1(T_0, T_r, \Delta P_1, \dots, \Delta P_n), \\ \max & f_2(T_0, T_r, \Delta P_1, \dots, \Delta P_n), \\ \min & f_3(T_0, T_r, \Delta P_1, \dots, \Delta P_n), \\ \text{s.t.} & T_0, T_r \in [10, 50](^\circ\text{C}), \\ & \Delta P_1, \dots, \Delta P_n \in [-250, 250](\text{MPa}), \end{cases} \quad (2)$$

where the decision vectors  $(T_0, T_r, \Delta P_1, \dots, \Delta P_n)$  are composed of the initial  $T_0$  and refrigeration  $T_r$  temperatures (in  $^\circ\text{C}$ ), and a discrete set of values denoted as  $\Delta P_1, \dots, \Delta P_n$  indicating the pressure variations (in MPa) applied to the High-Pressure equipment during the treatment. The objective functions are the averages of the activities in the food domain  $\Omega_F$  at the final time  $t_f$  computed as follows:

$$f_k(T_0, T_r, \Delta P_1, \dots, \Delta P_n) = \frac{1}{|\Omega_F|} \iint_{\Omega_F} A_k(r, z, t_f) dr dz,$$

where  $k = 1$  refers to the enzyme activity and  $k = 2$  to the vitamin activity; and the maximum temperature registered in the food sample  $\Omega_F$  during the processing time interval  $[t_0, t_f]$ :

$$f_3(T_0, T_r, \Delta P_1, \dots, \Delta P_n) = \max_{(r,z) \in \Omega_F, t \in [t_0, t_f]} T(r, z, t).$$

Evaluating these objective functions requires the numerical simulation of the HPT treatment using the heat transfer model for describing the pressure and temperature evolution. In order to reduce the complexity of the model, as proposed in (Infante et al. 2009), the convective terms have been neglected and an axisymmetrical configuration of the domain has been assumed leading to a two-dimensional formulation. For a detailed explanation about the resulting system of equations and the computational domain, see (Ferrández et al. 2017).

## 3 The original optimization algorithm

The Weighting Achievement Scalarizing Function Genetic Algorithm (Ruiz, Saborido, and Luque 2015), shortly called WASF-GA, is a multi-objective evolutionary optimization algorithm based on a population of points, named as individuals, to which some genetic operators inspired by the Darwin theory of

evolution are applied. In this way, it consists of an iterative procedure where, at each step, a new offspring population is generated from the previous-step population using these genetic operators. Then, among the individuals belonging to both populations, the ones which more accurately approach the Pareto optimal front, will survive composing the next-step population. Since WASF-GA is, in addition, a preference-based algorithm, this survival selection is also performed according to the closeness to certain predetermined values expressing the preferences of the person solving the problem. Consequently, the resulting set of solutions is an approximation to a part of the Pareto optimal front referred to as the region of interest.

In addition to the multi-objective function  $\mathbf{f} = (f_1, \dots, f_m)$ , the input parameters needed by WASF-GA are the following ones: (i) the size  $N$  of the population of individuals, (ii) the total number  $h_{\max}$  of iterations to be considered in the evolutionary procedure, (iii) the values  $p_m, d_m, p_c, d_c$  for the probabilities and the distributions of the genetic operators (mutation and crossover), (iv) a reference point  $\mathbf{q} = (q_1, \dots, q_m)$  giving the preferred values for the  $m$  objective functions, and (iv) a sample of  $N_\mu$  weight vectors  $W = \{\mu^1, \dots, \mu^{N_\mu}\}$ .

In WASF-GA, as usual in this kind of meta-heuristic algorithms, the initial population of  $N$  individuals is randomly generated in the search space. Then, at each iteration, three main stages can be distinguished:

1. *Reproduction*: The individuals of the previous-step population (called *parent population*) are matched between them for applying the crossover operator by pairs generating new individuals, which are later modified by the mutation providing the *offspring population*. Then, the *joint population* is built by the union of both parent and offspring populations.
2. *Classification*: Next, the classification is performed separating the individuals from this joint population into several groups called fronts. To do that, in WASF-GA, the weighted distances of the individuals to the reference point  $\mathbf{q}$  are computed using the  $L_\infty$  norm for each of the weight vector in the sample according to the Wierzbicki's Achievement Scalarizing Function (ASF) (Ruiz, Saborido, and Luque 2015). Then, each front is filled as follows. First, among the unclassified individuals, the one providing the lowest value for the ASF with the first weight vector is chosen to be the first individual of the front. After that, if there are still any unclassified individuals, among them, the one with the lowest value for the ASF considering the second weight vector is now selected and copied into the front. This process is repeated until all the individuals are classified or until the front contains exactly as many individuals as weight vectors in the sample,  $N_\mu$ . When the latter occurs and there are still individuals without classifying, a new front is created and filled with the same procedure.
3. *Selection*: The selection consists of building the parent population for the next iteration or for the outcome (if the number of iterations is equal to  $h_{\max}$ ). This new or final population is formed by the individuals of the first

fronts until reaching the total size of  $N$  individuals. If there is any front that cannot be included completely, then its individuals having the lowest ASF values are chosen. In our work, we assume that the population and the weight vector sample have the same size:  $N = N_\mu$ . Therefore, the new population built at the end of each iteration is composed exactly of the individuals in the first front.

## 4 The proposed improvements

This section is dedicated to explaining some ideas and techniques that have been added to the WASF-GA implementation in order to improve its performance when considering a low number of iterations.

We have distinguished between two categories depending on if the mechanisms affect the reproduction phase or the classification stage. Notice that the original selection procedure of WASF-GA has been maintained without any changes, i.e., as explained in Section 3. For the sake of completeness and usability of the notation, the original reproduction and classification methods have also been included in the enumerations at the first position.

In addition to the novelties applied in the reproduction and classification stages, we have introduced an external list with the aim of increasing the number of non-dominated points at the final population obtained with a short number of iterations. This external list is created at the beginning of the iterative process containing only the non-dominated points of the initial population. Then, each time a new population is generated, the external list is updated to include those new individuals which are non-dominated and remove those old individuals which are dominated by the new ones.

### 4.1 On the reproduction stage

At this stage, four alternatives have been considered: the WASF-GA reproduction method and three variations involving different techniques. The main difference among these alternative methodologies is the composition of the population named *joint population*. Notice that this population is created at the end of the reproduction phase and that the individuals included in it will be the ones that the algorithm will consider for the next classification and selection procedures.

- *Reproduction<sub>1</sub>*: The joint population is the union of the parent population and the offspring population created by applying crossover and mutation.
- *Reproduction<sub>2</sub>*: In addition to the parent and offspring populations, the *joint population* also contains two more populations:
  - The population of individuals obtained by improving the dominated solutions.

For this strategy, in addition to the external list, another list is built including those points which have not been included in the external list or which have been deleted from it because they are dominated. Then, for each one of those dominated points:

1. We look for the non-dominated point in the external list that is closest to it in the objective space using the Euclidean distance.
  2. We compute a new point randomly chosen in the segment that unites them in the search space.
- The secant population.

This mechanism is inspired by a similar technique that was successfully applied in a multilayer algorithm in (Ivorra, Mohammadi, and Ramos 2015) for accelerating the convergence of genetic algorithms.

The main idea is to generate at each iteration a new set of individuals called *secant population* employing the populations obtained at the two previous iterations. As explained in Section 3, the WASF-GA population can be thought of as a set of points, each one solving the minimization of a mono-objective function which is the ASF with a determined weight vector. Then, at each iteration  $h < h_{\max}$ , for each weight vector  $\mu^j = (\mu_1^j, \dots, \mu_m^j) \in W$ , with  $j = 1, \dots, N_\mu$ , the algorithm tries to obtain a point closer to the root of the ASF weighted using  $\mu^j$  and denoted as  $s^j(\mathbf{x})$ . Since the secant method is aimed at numerically approaching the zeros of a function without using any information about its gradient, we apply it to obtain a set of new individuals  $\{\mathbf{x}^{h,j}\}_{j=1,\dots,N_\mu}$  as follows:

$$\mathbf{x}^{h,j} = \text{proj}_S \left( \mathbf{x}^{h-1,j} - s^j(\mathbf{x}^{h-1,j}) \frac{\mathbf{x}^{h-1,j} - \mathbf{x}^{h-2,j}}{s^j(\mathbf{x}^{h-1,j}) - s^j(\mathbf{x}^{h-2,j})} \right),$$

where  $\mathbf{x}^{h-1,j}$  and  $\mathbf{x}^{h-2,j}$  are the individuals belonging to the previous-steps populations  $P^{h-1}$  and  $P^{h-2}$  because of their ASF values  $s^j(\mathbf{x}^{h-1,j})$  and  $s^j(\mathbf{x}^{h-2,j})$ , respectively. Notice that  $\text{proj}_S$  is the projection of the point into the search space  $S$  for guaranteeing that the new point is inside  $S$ .

- *Reproduction<sub>3</sub>*: As stated in literature, the local search procedures are aimed at improving the convergence to the optimum. In this case, we have used the multi-objective version of the SASS method called MOSASS, which was proposed in Redondo, Fernández, and Ortigosa (2017). Among its advantages, it must be highlighted that during the search process many interesting points are usually generated close to the individual to which the method is being applied. These points will be introduced in the external list of non-dominated points or in a new list containing the dominated individuals. Nevertheless, as a counterpart, this implies an increase in the number of evaluations being consumed. Then, we decided to apply MOSASS only to the individual of the current population that is the

closest one to the reference point in the objective space with the Euclidean distance.

MOSASS is based on an iterative procedure aimed at obtaining an improved point starting from an initial one. In addition to this initial point, it receives as input the normalized maximum radius  $\sigma_{ub} \in \mathbb{R}$  allowed for modifying the variables to avoid the generation of new points outside the region of interest. In our proposal, the value of  $\sigma_{ub}$  is the normalized distance between the initial point and the point of the parent population being the furthest from it in the search space. There are other parameters that are fixed to pre-determined values as the maximum number of iterations  $ic_{\max}$  and of consecutive failures  $Maxfcnt$  involved at the MOSASS stopping rule, the threshold number of consecutive failure trials  $F_{cnt}$  and of consecutive successful trials  $S_{cnt}$ , and the contraction  $ct$  and expansion  $ex$  coefficients (see Section 5.1).

In MOSASS, at each iteration, the starting individual  $\mathbf{x}$  is perturbed adding a multivariate Gaussian random vector  $\xi = N(b, \sigma I) \in \mathbb{R}^n$ , where  $\mathbf{b} \in \mathbb{R}^n$  is a normalized bias vector directing the search and  $\sigma \in \mathbb{R}$  is the radius of the most preferred search region. At the beginning of the iterative procedure,  $\mathbf{b} = \mathbf{0}$  and  $\sigma = \sigma_{ub}$ . Then, the new point  $\mathbf{x} + \xi$  is evaluated and, if it dominates the original individual  $\mathbf{x}$ , it replaces  $\mathbf{x}$  for the next iteration. On the contrary, we try to include  $\mathbf{x} + \xi$  in the external list but it will be accepted in this list only if it is not dominated by any individual on the list, otherwise, it will be transferred into the dominated list. This dominated list also receives the individuals that may be removed from the external list because they are dominated by the new individual  $\mathbf{x} + \xi$ . If  $\mathbf{x} + \xi$  cannot be stored in the external list because it is dominated, then we try the alternative point  $\mathbf{x} - \xi$  in the same way. First, we see if it dominates  $\mathbf{x}$  and, if not, we try to include it on the external list. At each iteration, a new Gaussian random vector is generated but updating the deviation  $\sigma$  that specifies the size of the sphere that most likely contains the perturbation vector, and the bias term  $\mathbf{b}$  indicating the center of this sphere. They are updated according to the number of successful trials achieved at the previous iterations. A trial point is considered successful if it dominates the previous point or if it is finally included in the non-dominated external list. If more than  $S_{cnt}$  consecutive successes have been reached, the perturbation radius is increased  $\sigma = ex \cdot \sigma$ . On the contrary, if more than  $F_{cnt}$  consecutive failures have occurred, it is decreased  $\sigma = ct \cdot \sigma$ . Furthermore, if the perturbation is too small or too big, it will be reset to  $\sigma^{ub}$ . In the first case, the bias vector is also reset to zero. For more details about MOSASS, see Lancinskas, Ortigosa, and Žilinskas (2013) and Redondo, Fernández, and Ortigosa (2017).

Finally, in *Reproduction*<sub>3</sub>, the joint population is formed by the offspring population, the individuals belonging to the external list, the dominated individuals and the individual obtained by the above-explained local search procedure.



- *Reproduction<sub>4</sub>*: We propose an alternative method for generating the offspring population consisting of building an *advanced population*. This advanced population is generated using the parent population, i.e. the population of the previous step, and the external list (see Algorithm 1). For each weight vector  $\mu^j$ , we select the point  $\mathbf{x}_{\text{el}}^j$  of the external list which minimizes the ASF weighted using  $\mu^j$  and, analogously, the point  $\mathbf{x}^j$  of the parent population minimizing the same ASF with  $\mu^j$ . Then, we randomly generate a new point  $\mathbf{x}_{\text{ad}}^j$  in the neighborhood between both points  $\mathbf{x}_{\text{el}}^j$  and  $\mathbf{x}^j$  as follows:

$$\mathbf{x}_{\text{ad}}^j = \mathbf{x}_{\text{el}}^j + \lambda(\mathbf{x}_{\text{el}}^j - \mathbf{x}^j), \quad (3)$$

where  $\lambda = (\lambda_1, \dots, \lambda_n)$  is a vector of random numbers. Next, we include this new point  $\mathbf{x}_{\text{ad}}^j$  in the advanced population. The idea is to achieve new points closer to those non-dominated points in order for them to have more probabilities to be non-dominated also. Sometimes it maybe be more successful to maintain some of the decision variables of the external list point because modifying all of them can lead to an objective value far from the neighborhood, especially when the number of variables is high. Thus, we have considered a probability value  $p_{\text{ad}} \in (0, 1]$ , which is pre-determined in advance and serves to make the decision of when to apply the modification to a certain variable (Lancinskas, Ortigosa, and Žilinskas 2013). More precisely, for each variable  $x_i, i \in \{1, \dots, n\}$ , a random number  $rnd$  is generated and, if  $rnd \leq p_{\text{ad}}$ , then the expression (3) is applied, otherwise,  $x_i = x_{\text{el},i}$  (see Algorithm 2).

---

**Algorithm 1** *calculateAdvancedPop*


---

```

1: calculateAdvancedPop           in:  population,  externalList
   out: advancedPop
2: IndividualSet advancedPop;
3: for  $j = 0$  to  $\text{problem.getNumberOfWeightVectors}()$  do
4:   double  $\text{ASFmin} = \text{ASF.evaluate}(\text{externalList.get}(0), \mu^j)$ ;
5:   int  $\text{indexASFmin} = 0$ ;
6:   for  $l = 0$  to  $\text{externalList.size}()$  do
7:     aux =  $\text{ASF.evaluate}(\text{externalList.get}(l), \mu^j)$ ;
8:     if  $\text{aux} < \text{ASFmin}$  then
9:        $\text{indexASFmin} = l$ ;
10:    end if
11:  end for
12:   $\text{popIndiv} = \text{population.get}(j)$ ;
13:   $\text{externalListIndiv} = \text{externalList.get}(\text{indexASFmin})$ ;
14:   $\text{advancedIndividual} = \text{calculateAdvancedIndividual}(\text{popIndiv},$ 
     $\text{externalListIndiv})$  ;
15:   $\text{advancedPop.add}(\text{advancedIndividual})$ ;
16: end for
17:  $\text{evaluatePopulation}(\text{advancedPop})$ ;
18: return advancedPop

```

---

---

**Algorithm 2** *calculateAdvancedIndividual*

---

```
1: calculateAdvancedIndividual in: popIndividual,  
   externalListIndividual out: advancedIndividual  
2: Individual advancedIndividual;  
3: double probab = 1/problem.getNumberOfVariables();  
4: double x0, x1, x2, lambda;  
5: boolean isthesame = true;  
6: while isthesame == true do  
7:   for i = 0 to problem.getNumberOfVariables() do  
8:     if rnd.nextDouble() <= probab then  
9:       x0 = popIndividual.getVariableValue(i);  
10:      x1 = externalListIndividual.getVariableValue(i);  
11:      lambda = rnd.nextDouble();  
12:      x2 = x1 + lambda * (x1 - x0);  
13:      x2 = Double.max(x2, getLowerBound(i));  
14:      x2 = Double.min(x2, getUpperBound(i));  
15:      advancedIndividual.setVariableValue(i, x2);  
16:      isthesame = false;  
17:     end if  
18:   end for  
19: end while  
20: return advancedIndividual
```

---

In order to maintain a good compromise between exploitation of the already known non-dominated points and exploration of the search space, we alternate the advanced individuals generation and the generation of the offspring population by means of the crossover and mutation operators, using the former at the even iterations and the latter at the odd iterations. Notice that, as the advanced population has as many individuals as weight vectors  $N_\mu$  and we have fixed  $N_\mu = N$ , the number of evaluations does not increment, i.e. it maintains the same number of evaluations per iteration than WASF-GA. It is very important for problems such as the one concerning us, in which the evaluation of the objective function has a high computational cost.

## 4.2 On applying the classification

The main core of the WASF-GA classification procedure detailed in Section 3 remains for all the studied versions. The only distinctive feature that we have modified is the choice of the population to which the classification will be applied.

- *Classification<sub>A</sub>*: As in the original WASF-GA, those individuals belonging to the joint population will be considered for their classification into fronts.

- *Classification<sub>B</sub>*: When the external list has enough individuals, i.e. more than the number of required individuals in the final population, only these individuals belonging to the external list are considered for the classification and selection procedures. As a consequence, we guarantee that the final population is formed only by non-dominated individuals.

## 5 Computational experiments

According to Section 4, the different variants of WASF-GA that have been implemented and tested are denoted as WASF-GA<sub>rc</sub>, where the sub-index  $r = 1, 2, 3, 4$  refers to the considered reproduction method and  $c = A, B$  indicates one of the proposed classification options. Hence, the original WASF-GA corresponds to WASF-GA<sub>1A</sub>. To compare their solutions, the execution of each variant WASF-GA<sub>rc</sub> has been performed 10 times considering the food processing problem (2). Then, the quality of the obtained Pareto front approximations has been analyzed by using the average and the standard deviation values of some state-of-the-art effectiveness measures.

The experiments were performed on a cluster node which has 8 Intel Xeon E7 8860v3 3.2 GHz processors with 16 cores each and 2.3 TB of RAM. The shared-memory parallel version of WASF-GA proposed in (Ferrández et al. 2018) has been used adapting it to hold all these WASF-GA<sub>rc</sub> variants.

As said in Section 2, for the considered industrial problem, evaluating the objective functions implies the numerical simulation of the HPT treatment. Despite considering the simplifications also detailed in Section 2, the computational cost of approximating the model using the Finite Element Method (FEM) is quite high. Looking for a compromise between computational cost and accuracy of the solution, a spatial mesh with 2881 elements has been considered for the discretized domain. The computational time of a single evaluation is 46.5 seconds in average using this mesh in the cluster node described before with a single core per evaluation. Considering a mesh with a lower number of points may lead to a poor description of the HPT treatment not being able to capture the effects of the pressure and temperature over the micro-organism activities.

### 5.1 Optimization settings

According to previous works in which we dealt with this industrial problem (Ferrández et al. 2017; Ferrández et al. 2018), we need a population size of at least  $N = 200$  individuals for the optimization algorithm. Considering fewer individuals in the population, the obtained set of optimal points conforming the Pareto front approximation may not sufficiently cover the region of interest. As a consequence, it cannot be guaranteed that some demanding quality scenarios for the vitamin conservation and the enzyme reduction will be attended.

Concerning the maximum number of iterations, the higher its value, the higher the quality of the Pareto front approximation in both senses, the better distributed the points and the closer they are from the real Pareto optimal

front. However, in view of the required amount of individuals and the high computational cost of their evaluation, to solve our industrial problem in an affordable time, we have considered a maximum number of iterations of  $h_{\max} = 50$ .

The stopping criterion of the original algorithm WASF-GA<sub>1A</sub> is based on the number of iterations. More precisely, its evolutionary process is running until the value  $h_{\max}$  is reached. Since WASF-GA<sub>1A</sub> consumes as many evaluations per iteration as individuals in the population, when it finishes, the total number of evaluations done is  $N \cdot h_{\max}$ . Nevertheless, for some of the proposed variants of WASF-GA<sub>1A</sub>, the number of evaluations at each iteration is not a constant number. In particular, it occurs for WASF-GA<sub>2c</sub>, whose reproduction method involves the improvement of the individuals that are dominated, and for WASF-GA<sub>3c</sub>, which is based on the use of the local search technique MOSASS, no matter the classification method  $c = A$  or  $c = B$  used. Thus, for the sake of the fairness of their comparison, the stopping rule described on Algorithm 3, which is based on the number of evaluations *cont\_eval*, has been implemented.

---

**Algorithm 3** *isStoppingConditionReached*

---

```

1: isStoppingConditionReached   in: int cont_eval, int N, int hmax
   out: boolean stop
2: boolean stop = false;
3: if cont_eval  $\geq N \cdot h_{\max}$  then
4:   stop = true;
5: else if cont_eval + N  $\geq N \cdot h_{\max}$  then
6:   stop = true;
7: else
8:   stop = false;
9: end if
10: return stop

```

---

Next, we include the used settings concerning the remaining parameters whose fine-tune analysis falls out of the scope of this paper but for which the most extended values used in literature have been selected (Solis and Wets 1981; Ruiz, Saborido, and Luque 2015). Regarding the genetic operators, the Simulated Binary Crossover (SBX) and the polynomial mutation have been employed with probabilities  $p_c = 0.9$  and  $p_m = 0.1$ , respectively, and a distribution of 20 for both. In particular, for those parameters, we performed a more exhaustive study and those values proved to be the ones leading to the higher quality Pareto front approximations (Ferrández et al. 2017).

The MOSASS parameters used in the implementation of WASF-GA<sub>3c</sub> (with  $c = A, B$ ) are the ones recommended by Solis and Wets (1981) and successfully probed by Lancinskas, Ortigosa, and Žilinskas (2013) and Redondo, Fernández, and Ortigosa (2017). The coefficient values used for updating the bias term  $\mathbf{b}$  are 0.4, 0.2 and 0.5, and the values for the threshold number of successes and failures involved at the  $\sigma$  computation are  $Scnt = 5$  and  $Fcnt = 3$ , respec-

tively. For the contraction and expansion coefficients, also the recommended values  $ct = 0.5$  and  $ex = 2$ , respectively, are considered. In order to avoid an excessive increment of the number of evaluations derived from the application of the MOSASS mechanism, the maximum number of consecutive failures for its stopping rule has been fixed to  $Maxfent = 5$  for each WASF-GA<sub>3c</sub> iteration. In both methods, MOSASS in WASF-GA<sub>3c</sub> and the generation of the advanced individuals in WASF-GA<sub>4c</sub>, a probability  $p_{ad} = 1/n$ , where  $n$  is the number of variables, is considered for deciding when the decision variables are modified or not (Lancinskas, Ortigosa, and Žilinskas 2013).

## 5.2 Quality indicators

To measure the effectiveness of the algorithms, we have used the most extended methodology in literature, which is the *quality indicator* procedure. It consists of quantifying the quality of each Pareto front approximation and, then, analyzing the resulting numbers for their comparison.

As the studied algorithms are heuristics, every particular instance has been run 10 times. Thus, for each algorithm  $rc$ , with  $r = 1, \dots, 4$  and  $c = A, B$ , ten different Pareto set approximations  $PS_{rc}^1, \dots, PS_{rc}^{10}$  in the decision space have been obtained. All these resulting sets for all the algorithms compose the set of all the Pareto-set approximations denoted as  $SPS$ . Since the real Pareto front is required for the computation of some quality indicators, in problems like ours where it is not known, a reference set  $RS$  is used instead for approximating it. In our work,  $RS$  has been generated by merging all the individuals of the  $SPS$  Pareto-set approximations, selecting those of them which are non-dominated and obtaining their image in the objective space. Furthermore, for a fairness contribution of all the objectives,  $RS$  and all the Pareto front approximations have been normalized before computing the quality indicator values (Ferrández et al. 2017). The following standard normalization has been used:

$$f_k(\mathbf{x})' = \frac{f_k(\mathbf{x}) - f_k^{(\min)}}{f_k^{(\max)} - f_k^{(\min)}}, \quad k = 1, \dots, m,$$

where  $f_k^{(\min)}$  (resp.  $f_k^{(\max)}$ ) denotes the minimum (resp. maximum) value of  $f_k$  when considering all the solutions in  $SPS$ .

In general, there exist three kinds of quality indicators depending on what feature they measure: proximity, diversity and global indicators. The former are focused on computing the distance between the real Pareto front (or, in this case, the  $RS$ ) and the approximation obtained with the optimization algorithm. Therefore, since our main goal is comparing the algorithms in terms of their convergence, we have used some proximity indicators. In particular, we have employed two of those measures, namely, the *additive epsilon* (Zitzler et al. 2003) and the *inverted generational distance plus* indicator (IGD+) (Ishibuchi et al. 2015). Additionally, to evaluate the global quality, we have used the well-known *hypervolume* indicator based on computing the hypervolume of the piece of the

decision space which is weakly dominated by the Pareto front approximation (While, Bradstreet, and Barone 2012; Zitzler and Thiele 1998).

On the one hand, given a Pareto set approximation  $PS$ , the additive epsilon indicator is calculated as the minimum distance by which  $\mathbf{f}(PS)$  needs to be moved in each dimension of the objective space such that the reference set  $RS$  is weakly dominated. Mathematically, it can be expressed as follows:

$$I_{\epsilon+}(\mathbf{f}(PS)) = \min_{\epsilon \in \mathbb{R}} \left\{ \forall \mathbf{b} \in RS, \quad \exists \mathbf{a} \in PS : \right. \\ \left. \frac{f_k(\mathbf{a}) - f_k^{(\min)}}{f_k^{(\max)} - f_k^{(\min)}} - \epsilon \leq \frac{f_k(\mathbf{b}) - f_k^{(\min)}}{f_k^{(\max)} - f_k^{(\min)}}, \quad \forall k \in \{1, \dots, m\} \right\}. \quad (4)$$

On the other hand, the inverted generational distance metric plus (IGD+), which was recently proposed by Ishibuchi et al. (2015), uses the same formula as the inverted generational distance:

$$\text{IGD}+(\mathbf{f}(PS)) = \frac{1}{|RS|} \left( \sum_{l=1}^{|RS|} (d_l^+)^p \right)^{1/p}$$

where a new modified distance  $d^+$  is employed instead of the Euclidean distance for calculating the distance  $d_l^+$  from  $\mathbf{b}_l \in RS$  to its nearest objective vector in  $\mathbf{f}(PS)$ . For a minimization problem, this modified distance is defined as:

$$d^+(\mathbf{a}, \mathbf{b}) = \sqrt{(\max\{a_1 - b_1, 0\})^2 + \dots + (\max\{a_m - b_m, 0\})^2},$$

for two objective vectors  $\mathbf{a} \in PS$  and  $\mathbf{b} \in RS$ . In this work, we have used  $p = 2$ .

Notice that all those three indicators that we have employed are *Pareto compliant*. It means that whenever a Pareto-set approximation  $A$  is preferable to a Pareto-set approximation  $B$  with respect to weak Pareto dominance, the indicator value for  $\mathbf{f}(A)$  should be at least as good as the indicator value for  $\mathbf{f}(B)$ . To compute them, we have used the implementations available at the optimization package jMetal (Durillo and Nebro 2011).

## 6 Results and discussion

In Fig. 1, the average results obtained by the different algorithms for the epsilon and the IGD+ indicators computed every 10 iterations of the evolutionary procedure, are given. Notice that for these metrics the lower the value, the better the quality. Therefore, the versions WASF-GA<sub>3A</sub> and WASF-GA<sub>4B</sub> based on the introduction of the MOSASS technique and the advanced population with classification applied to the external list, respectively, clearly outperform the original algorithm WASF-GA<sub>1A</sub>. It means that using those mechanisms to solve the considered industrial problem, we achieve to enhance the convergence even when the number of iterations is low.

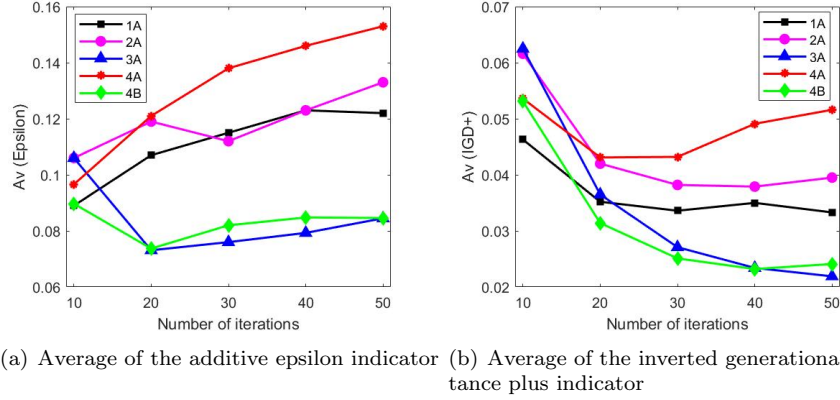


Figure 1: Evolution of the average of the additive epsilon and the inverted generational distance plus indicators computed for  $h = 10, 20, 30, 40, 50$  iterations. 1A corresponds to the original WASF-GA, 2A to its version based on improving the dominated individuals and building a secant population, 3A to the one using the MOSASS mechanism, 4A to the one considering the advanced population, and 4B the algorithm also using the advanced population but only considering the points of the external list for the classification procedure.

For the hypervolume global indicator, the higher the value, the better the solution. Thus, as can be seen in Fig. 2(a), the algorithm WASF-GA<sub>4B</sub> using the advanced population for the reproduction phase and the external list for the classification procedure, also overcomes the basic version WASF-GA<sub>1A</sub> in average.

According to those results for the considered industrial problem, we have achieved our goals of improving WASF-GA in terms of convergence and proximity to the reference Pareto front but also in global performance. Now, in previous studies, we realized that the population obtained with WASF-GA does not reach the total number of individuals being non-dominated, in our case  $N = 200$ . This was the reason for introducing the variant denoted as *Classification<sub>B</sub>* which only considers the individuals belonging to the external list for the classification into fronts. As a consequence, all the individuals obtained as outcome of WASF-GA<sub>4B</sub> are non-dominated. As can be seen in Fig. 2(b), in comparison to the other considered versions, it supposes a great improvement. In our problem, it implies that the solution set provides the food engineer a larger amount of temperature and pressure configurations for the HPT equipment allowing to attend more quality demands maybe from different countries.

In Table 1, the average and standard deviation values of the quality indicators for the maximum number of iterations considered  $h_{max} = 50$ , are shown. The best values for each of the indicators have been highlighted in bold. It is important to mention that, according to these values, the algorithms WASF-GA<sub>3A</sub> and WASF-GA<sub>4B</sub> are not only better in average than the others for the

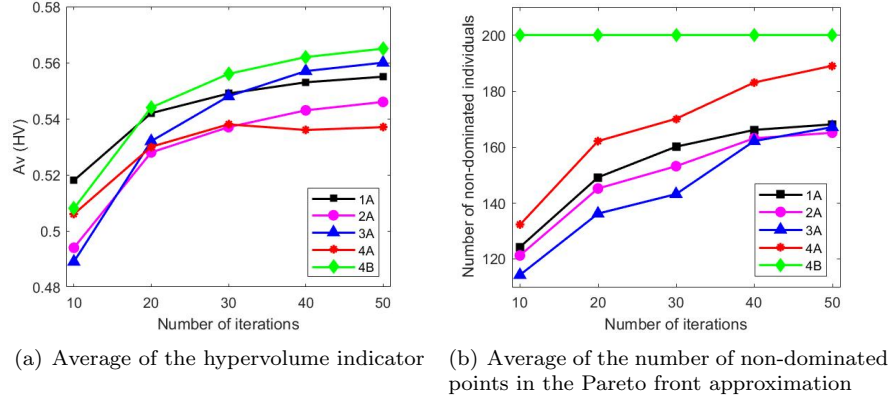


Figure 2: Evolution of the average of the hypervolume and the number of non-dominated points computed for  $h = 10, 20, 30, 40, 50$  iterations. 1A corresponds to the original WASF-GA, 2A to its version based on improving the dominated individuals and building a secant population, 3A to the one using the MOSASS mechanism, 4A to the one considering the advanced population, and 4B the algorithm also using the advanced population but only considering the points of the external list for the classification procedure.

epsilon, IGD+ and hypervolume indicators but they also seem to have a low variability of the results since their standard deviation values are lower. As a consequence, even the execution of WASF-GA<sub>1A</sub> providing the better indicator values seems to have an inferior quality in comparison to the execution of WASF-GA<sub>4B</sub> giving its worst results.

To sum up, in general, the proposed algorithm WASF-GA<sub>4B</sub> obtains better results for the food processing problem than the other considered versions for all the analyzed measures. Moreover, it guarantees that all the individuals in the outcome population are non-dominated solutions.

## 7 Conclusions

In this work, some mechanisms improving the convergence of evolutionary multi-objective optimization algorithms have been analyzed and tested to solve a complex industrial problem. Some of them are inspired by successful techniques found in the literature as the secant method or the MOSASS local search. Another, such as the one based on the construction of an advanced population, has been completely designed by the authors.

More precisely, they have been applied to enhance the performance of the preference-based algorithm known as WASF-GA when a low number of iterations is considered. The different WASF-GA versions have been tested to solve a particular industrial problem consisting of optimizing the High-Pressure and



Table 1: Average and standard deviation values of the considered quality indicators for  $h_{\max} = 50$  iterations. *1A* corresponds to the original WASF-GA, *2A* to its version based on improving the dominated individuals and building a secant population, *3A* to the one using the MOSASS mechanism, *4A* to the one considering the advanced population, and *4B* the algorithm also using the advanced population but only considering the points of the external list for the classification procedure. The best values for each of the indicators have been highlighted in bold.

<i>rc</i>	Epsilon		IGD+	
	Av	Dev	Av	Dev
<i>1A</i>	1.22E-01	2.9E-02	3.33E-02	1.2E-02
<i>2A</i>	1.33E-01	5.4E-02	3.95E-02	2.1E-02
<i>3A</i>	<b>8.45E-02</b>	<b>4.9E-03</b>	<b>2.19E-02</b>	<b>1.2E-03</b>
<i>4A</i>	1.53E-01	3.6E-02	5.16E-02	1.6E-02
<i>4B</i>	<b>8.46E-02</b>	<b>3.8E-03</b>	<b>2.41E-02</b>	<b>5.7E-03</b>

<i>rc</i>	HV		Number of ND	
	Av	Dev	Av	Dev
<i>1A</i>	5.55E-01	8.9E-03	1.68E+02	6.6E+00
<i>2A</i>	5.46E-01	1.6E-02	1.65E+02	4.9E+00
<i>3A</i>	5.60E-01	9.6E-04	1.67E+02	8.6E+00
<i>4A</i>	5.37E-01	1.5E-02	1.89E+02	7.0E+00
<i>4B</i>	<b>5.65E-01</b>	<b>6.1E-03</b>	<b>2.00E+02</b>	<b>0.0E+00</b>

Temperature food processing treatment. Then, an exhaustive analysis of their results using some measures of the Pareto front quality has been carried out. From these quality results, we can conclude that:

- Implementing the creation of an advanced population, the obtained Pareto front approximations show better values than the ones of the original WASF-GA for all the considered proximity and global quality indicators. This advanced population method allows us to generate new points closer to the non-dominated individuals of the external list, such that they have more probabilities of also being non-dominated.
- Introducing an external list for storing the non-dominated individuals found during the iterative process and using this list for the classification stage, we achieve that all the points in the final population are non-dominated.

As a future work, we will attempt to also improve some quality measures based on the distribution of the points in the Pareto front. Furthermore, we will explore the use of surrogate models. These kinds of models, which are becoming more popular over the last few years, are also prescribed for problems whose evaluation is expensive and involves complex simulations.

## Acknowledgments

This research has been funded by grants from the Spanish Ministry of Economy and Competitiveness (TIN2015-66680-C2-1-R and MTM2015-64865P); Junta de Andalucía (P12-TIC301), in part financed by the European Regional Development Fund (ERDF). Juana López Redondo is a fellow of the Spanish “Ramón y Cajal” contract program, co-financed by the European Social Fund. The work has been performed under the Project HPC-EUROPA3 (INFRAIA-2016-1-730897), with the support of the EC Research Innovation Action under the H2020 Programme; in particular, the author gratefully acknowledges the support of Ben Paechter from the School of Computing at Edinburgh Napier University and the computer resources and technical support provided by EPCC (Edinburgh Parallel Computing Center). This work used EPCC’s Cirrus HPC Service (<https://www.epcc.ed.ac.uk/cirrus>).

## References

- Durillo, J. J., and A. J. Nebro. 2011. “jMetal: A Java framework for multi-objective optimization.” *Advances in Engineering Software* 42: 760–771.
- Ferrández, M. R., S. Puertas-Martín, J. L. Redondo, B. Ivorra, A. M. Ramos, and P. M. Ortigosa. 2018. “High-performance computing for the optimization of high-pressure thermal treatments in food industry.” *The Journal of Supercomputing* 1–16.

- Ferrández, M. R., J. L. Redondo, B. Ivorra, A. M. Ramos, and P. M. Ortigosa. 2017. “A Decision Tool based on a Multi-Objective Methodology for designing High-Pressure Thermal Treatments in Food Industry.” Preprint version.
- Filatovas, E., A. Lančinskas, O. Kurasova, and J. Žilinskas. 2016. “A preference-based multi-objective evolutionary algorithm R-NSGA-II with stochastic local search.” *Central European Journal of Operations Research* 1–20.
- Gomez, S., B. Ivorra, and A. M. Ramos. 2011. “Optimization of a pumping ship trajectory to clean oil contamination in the open sea.” *Mathematical and Computer Modelling* 54 (1): 477–489.
- Infante, J. A., B. Ivorra, A. M. Ramos, and J. M. Rey. 2009. “On the Modelling and Simulation of High Pressure Processes and Inactivation of Enzymes in Food Engineering.” *Mathematical Models and Methods in Applied Sciences (M3AS)* 19 (12): 2203–2229.
- Ishibuchi, H., H. Masuda, Y. Tanigaki, and Y. Nojima. 2015. “Modified distance calculation in generational distance and inverted generational distance.” In *International Conference on Evolutionary Multi-Criterion Optimization*, 110–125. Springer.
- Ivorra, B., B. Mohammadi, and A. M. Ramos. 2015. “A multi-layer line search method to improve the initialization of optimization algorithms.” *European Journal of Operational Research* 247 (3): 711–720.
- Ivorra, B., J. L. Redondo, J. G. Santiago, P. M. Ortigosa, and A. M. Ramos. 2013. “Two-and three-dimensional modeling and optimization applied to the design of a fast hydrodynamic focusing microfluidic mixer for protein folding.” *Physics of fluids* 25 (3): 032001.
- Lancinskas, A., P. M. Ortigosa, and J. Žilinskas. 2013. “Multi-objective single agent stochastic search in non-dominated sorting genetic algorithm.” *Nonlinear Analysis: Modelling and Control* 18 (3): 293–313.
- Redondo, J. L., J. Fernández, and P. M. Ortigosa. 2017. “FEMOEA: a Fast and Efficient Multi-Objective Evolutionary Algorithm.” *Mathematical Methods of Operations Research* 85 (1): 113–135.
- Ruiz, A. B., R. Saborido, and M. Luque. 2015. “A preference-based evolutionary algorithm for multiobjective optimization: the weighting achievement scalarizing function genetic algorithm.” *Journal of Global Optimization* 62 (1): 101–129.
- Solis, F. J., and R. J. B. Wets. 1981. “Minimization by random search techniques.” *Mathematics of operations research* 6 (1): 19–30.
- While, L., L. Bradstreet, and L. Barone. 2012. “A fast way of calculating exact hypervolumes.” *IEEE Transactions on Evolutionary Computation* 16 (1): 86–95.

- Zitzler, E., and L. Thiele. 1998. "Multiobjective optimization using evolutionary algorithms - a comparative case study." In *Parallel problem solving from nature V*, edited by A. E. Eiben, Amsterdam, September, 292–301. Springer-Verlag.
- Zitzler, E., L. Thiele, M. Laumanns, C. M. Foneseca, and V.G. da Fonseca. 2003. "Performance assessment of multiobjective optimizers: An analysis and review." *IEEE Transactions on Evolutionary Computation* 7 (2): 117–132.